

Eindtoets T07351 Software engineering

Een eindtoets staat in het algemeen model voor het tentamen van de betreffende cursus. Aangezien deze cursus een mondeling tentamen heeft, bevat deze eindtoets slechts een aantal voorbeelden van het soort vragen dat op het tentamen gesteld wordt. Meestal wordt een bepaald onderwerp uit de cursus aangesneden en wordt daarover een serie vragen gesteld.

Vraag 1. Verificatie en validatie (V&V)

- a. Wat wordt verstaan onder V&V?
- b. Uit welke kernactiviteiten bestaat V&V?
- c. Noem enkele voor- en nadelen van inspecties.
- d. Idem voor testen.

Vraag 2. Verificatie en validatie (V&V)

- a. Wat is het verschil tussen testen en debugging?
- b. Wat is het verschil tussen black-box- en white-boxtesten?
- c. Wat kun je met testen aantonen en wat niet?

Vraag 3. Configuratiebeheer

- a. Wat wordt verstaan onder CM?
- b. Welke soorten producten vallen onder CM?
- c. Waarom is CM eigenlijk nodig?
- d. Noem enkele kernactiviteiten van CM.
- e. Is CM nodig nadat een systeem operationeel is?
- f. Wordt binnen XP ook CM toegepast?

Vraag 4. XP en RUP

- a. Noem enkele belangrijke principes en kenmerken van XP.
- b. Noem enkele belangrijke kenmerken RUP.
- c. Wat wordt binnen RUP gedaan aan testen?
- d. Wat wordt binnen XP gedaan aan testen?

Vraag 5. Software re-engineering

Noem een aantal activiteiten van het software re-engineering process.

Vraag 6. System models

- a. Beschrijf de rol van contextmodellen in het software engineering process.
- b. Noem enkele vormen van contextmodellen en hun specifieke eigenschappen.
- c. Beschrijf de rol van interactiemodellen in het software engineering process.
- d. Wat is de relatie tussen contextmodellen en interactiemodellen?

Vraag 7. Dependability

- a. Wat wordt verstaan onder 'dependability'?
- b. Noem 4 aspecten van dependability.

Vraag 8. Service-oriented software engineering

- a. Wat zijn de verschillen tussen een 'service' en een 'webservice'?
- b. Beschrijf 3 essentiële processtappen welke uitgevoerd moeten worden bij 'Service engineering'.
- c. Noem enkele problemen bij het testen van een systeem gebaseerd op services.

Vraag 9. Aspect-oriented software development

- a. Wat wordt bedoeld met 'separation of concerns'?
- b. Geef voorbeelden van 'viewpoints' en 'concerns'.
- c. Beschrijf wat een aspectweaver doet.

Uitwerkingen

Vraag 1. Verificatie en validatie (V&V):

- a. Zie pagina 207 van het tekstboek.
- b. Een globaal antwoord staat in paragraaf 2.2.3 op blz. 41 van het tekstboek. Meer details zijn te vinden in de inleiding van hoofdstuk 8 op blz. 206-210 van het tekstboek.
- c.
Voordelen:
 - toepasbaar op allerlei producten (documentatie, programmacode, ...),
 - vroeg in het ontwikkelingstraject toepasbaar.

Nadelen:

- moeilijk toepasbaar op een groter product,
 - moeilijk toepasbaar voor eigenschappen als gebruiksgemak en prestaties.
- d.
Voordelen:
 - toepasbaar op een werkend systeem(onderdeel),
 - toepasbaar op eigenschappen als gebruiksgemak en prestaties.

Nadelen:

- alleen toepasbaar op programmatuur,
- laat in het ontwikkelingstraject toepasbaar.

Vraag 2. Verificatie en validatie (V&V):

- a. Bij testen kunnen fouten optreden en met debugging kunnen die fouten worden opgespoord (en gerepareerd).
- b. Bij black-box testen worden testgevallen alleen op grond van de specificatie bepaald en bij white-box testen op grond van de inhoud en structuur van het betreffende onderdeel.
- c. Met testen kan alleen de aanwezigheid van fouten aangetoond worden, niet de afwezigheid. Anders gezegd: met testen kan alleen de incorrectheid van een systeem worden aangetoond, niet de correctheid.

Vraag 3. Configuratiebeheer:

- a. Het beheren van alle onderdelen van een softwareproduct in ontwikkeling of in onderhoud, met name de versies daarvan die ontstaan door wijziging, en ook van de samenhang tussen die onderdelen.
- b. Code, systeem- en projectdocumentatie, testverzamelingen en –resultaten, gebruikshandleidingen, change-requests, ...
- c. Vanwege de enorm grote aantallen configuratie-items, dus van de versies van alle onderdelen, en de relaties ertussen. Software engineers werken meestal in teamverband aan een product, versiebeheer is daarbij onontbeerlijk.
- d. Versie-, veranderings- en opleveringsbeheer.

- e. Dat is zeker nodig als het systeem nog in onderhoud blijft, zeker als dat intensief zal zijn of als verschillende versies van een systeem naast elkaar bestaan en in onderhoud blijven.
- f. XP heeft enkele kenmerken die CM noodzakelijk maken, bijvoorbeeld:
 - gemeenschappelijk codebezit, hetgeen inhoudt dat iedereen op elk moment elk deel van de code mag veranderen, en
 - doorlopende integratie, hetgeen inhoudt dat veranderingen aan onderdelen snel worden ingebracht in het systeem en dat het gehele systeem vaak opnieuw wordt gecompileerd.

Vraag 4. XP en RUP:

- a. Principles: zie bladzijde 65 van het tekstboek. Kenmerken of 'practices': zie bladzijde 66 van het tekstboek, figure 3.4.
- b. Zie paragraaf 2.4 van het tekstboek.
- c. Zie paragraaf 3.3.1 van het tekstboek.
- d. Zie bladzijde 52 van het tekstboek.

Zie in dit verband ook Opdracht 2. Geef in deze eindtoets ook uw eigen mening over de toepasbaarheid van de door u genoemde principes en methoden.

Vraag 5. Software re-engineering:

Zie tekstboek paragraaf 9.3.2, met name ook figure 9.11.

Vraag 6. System models:

- a. Contextmodellen leggen de grenzen van het systeem vast: wat hoort erbij en wat niet.
- b. Procesmodel, UML activity diagram, zie bladzijde 123 van het tekstboek.
- c. Zie bladzijde 124 van het tekstboek.
- d. In contextmodellen wordt de context of omgeving van het systeem vastgelegd, in interactiemodellen de interactie tussen het systeem en omgeving, of tussen componenten of deelsystemen.

Vraag 7. Dependability:

- a. Het woord 'dependability' heeft in dit verband geen vertaling in het Nederlands. Het beschrijft in hoeverre je je kunt 'verlaten' op het systeem. Het is dus meer dan alleen betrouwbaarheid.
- b. De 4 aspecten van dependability zijn:
 - betrouwbaarheid,
 - beschikbaarheid,
 - beveiliging,
 - veiligheid.

Zie ook figure 11.1 van het tekstboek en de bijbehorende tekst.

Vraag 8. Service-oriented software engineering:

- a. De verschillen tussen een ‘service’ en een ‘webservice’ zijn:
 - een service is een stuk functionaliteit, met een nauwkeurige beschrijving en een nauwkeurig beschreven interface,
 - een webservice werkt via internet met (standaard) web-protocollen.
- b. zie tekstboek figure 19.6, bladzijde 519.
- c. zie tekstboek paragraaf 19.3.2, bladzijde 533.

Vraag 9. Aspect-oriented software development:

- a. zie tekstboek paragraaf 21.1, bladzijde 567.
- b. Voor concerns: zie punt 1 t/m 5 op bladzijde 568 van het tekstboek.
Voor concerns en viewpoints: zie bijvoorbeeld figure 21.8 op bladzijde 578 van het tekstboek.
- c. zie tekstboek paragraaf 21.2, bladzijde 571 en verder.